Stephen Nemrow
I SYS 520
Final Project

Executive Summary:

My wife is employed by a wealthy individual with a large personal library. A large portion of the library consists of interior design and landscaping books, many of which are still in plastic wrap and have never been read. My wife's employer is considering donating these books to a local library and wants to know how much the books are worth so she can calculate the tax benefit of the donation. She has asked my wife to get this information for her. My wife has gathered over one thousand ISBN numbers and was dreading having to search each book and pull all of the information.

My program takes a list of ISBN numbers and searches amazon for each ISBN. It then scrapes Amazon for the title, new and used prices, author, published year, and book rating (on a scale of 1 to 5). It returns the result to excel and summarizes the value of the books, based on whether the book is new or used.

Implementation:

Running the Code

To simplify running the code I added a new ribbon tab named summarize. This tab contains 3 tabs the user can use to get the book info, get the summary information and reset the sheet to begin again.

The sheet itself is split into 3 sections. The top blue/grey area is the summary. This will return the value of the library as priced on Amazon. The number of errors is the number of ISBN for which there was either no pricing information, or multiple listings which the user needs to address. The light orange area is where the user enters the ISBN numbers and the books condition. The dark orange section is populated when the 'Get Book Info' button is pressed. That information is used to generate the summary. The 'Reset Sheet' will remove all of the values in the orange section and zero out the blue section.

	A	B	C	D	E	F	G
1							
2	**Total Value**	0.00					
3	**New Book Number**	0	**Used Book Number**	0	**Number of Errors**	0	
4	**New Book Value**	0.00	**Used Book Value**	0.00			
5							
6	**ISBN**	**Condition**	Title	Author(s)	Date Published	New Price	Used Pric
7	9781579653484	New					
8	9780891348160	Used					
9	9781554074525	New					
10	9780789306661	New					
11	1580112609	Used					
12	9781770850453	Used					
13	9781574863901	New					
14	9781574863918	Used					
15	9780696208577	New					
16	9781588166951	New					
17	9781588163172	Used					

Getting Book Info

1. All of the desired information can be found on the search results page. When a search is done for a specific ISBN, Amazon almost always returns a single result, which is the desired book. We first validate that the ISBN. Then we proceed.

2. Generate the search URL. The search URL is comprised of the base URL and the ISBN number. Example: The search for the ISBN of 9781479167050 is http://www.amazon.com/s/field-keywords=9781479167050.

3. Using the agent module we read in the HTML and parse it for the required information
First we determine if the search yielded any results. Figure 1 shows the possible outcomes of the search. By searching for unique characteristics within the HTML we can identify these outcomes. We look within the template div and identify the centerplus div. If there are no results there is an h1 with the id of noresultstitle.

Figure 1.

If there are results there is an h2 element with a class of resultcount. We then check to see if the number is greater than 1.

If there are no results or multiple results we put the link on the worksheet along with a status noting whether there were 0 or more than 1 result. We leave out the pricing information so that the user can investigate further.

4. Gather Title, Author and Year Published. When parsing the HTML for the title, author, and year published, we look within the HTML with an h3 element with a class of newaps. There are several contingencies to consider. Figure 2 demonstrates some of these contingencies.

Figure 2

These images show the different conditions. There is always a title, but the author name and published date are optional. The title is always within an anchor tag. We look for that tag and pull out the title. When the author name is presented, it may be presented within an anchor tag or not. It may be also grouped with other people's names, with the possibility that each name is in its own anchor tag. When a published date is presented it is always in between a pair of parenthesis, and we parse that information for just the year. The code accounts for all of these contingencies, and correctly extracts the data.

5. Gather price data and book type. The pricing information and book type can be found within a UL element with the class of rsltL. Within that element there is an LI element for each price. We loop through the li elements and extract the information. There are several configurations presented, although some are less likely to occur. Figure 3 demonstrates these configurations.

Figure 3

These images describe the different configurations. There is the possibility of the book being out of print or unavailable. If this occurs the program sets the status field to "No Price" and stops processing that ISBN. Next is the possibility that Amazon offers its own new price. When this occurs the price is not followed by the words "new" or "used". We inspect the words following the price and when it is not "new" or "used" then the value is the book type. If both the Amazon new price and another new price exist, the Amazon new price is used. If not, then the new price is used. If the value after the price is "used" then it is placed in the used price field. If there is no Amazon price then there is no book type available.

6. Gather book rating. The rating information is found within the UL element class="rsltR dkGrey". There are two possible configurations. When there is review information and when there isn't. Figure 4 shows these configurations.

Figure 4

When there is review information there is a LI element with class="rvw". Within that element there is an A tag with an alt element with the rating. We parse that alt element and get the first 3 characters for the rating. If there is no LI element with class="rvw" then there is no review available.

All of the gathered information is placed on the dark orange section of the spreadsheet. We have completed the iteration for the ISBN. We then continue on to the next item.

Learning

My initial attempt at this project was to program it using the Amazon Product Advertising API to gather the information about the books on the list. But I quickly ran into a problem. In order to query the Amazon API I was required to use sign each request using cryptographic techniques. While packages have been built for developers in Java as well as other languages I couldn't find one for Visual Basic. I searched the internet to find anyone who had previously implemented a working request signing sub procedure but found no real solution. I found a general guide to signing requests here. (http://docs.aws.amazon.com/AWSECommerceService/latest/DG/rest-signature.html). I followed the instructions but got stuck on calculating a RFC 2104-compliant HMAC with the SHA256 hash algorithm. All signatures I generated were not recognized by Amazon. I then attempted to use a javascript library using the Jscript reference. I was able to get the code to generate a signature, but not one that Amazon would accept. After spending about 20 hours with this problem, in the end I decided to move on and use the agent module and scrape the Amazon search listings.

When using the agent module I found that I needed to keep track of where I was at all times, and check conditions before proceeding. I often would end up looking for text that existed on one page, but not on another. I was required to look at many versions of similar pages to identify the truly unique parts of each page.

Assistance

The only assistance I received was using the code we did in class validating a 10 digit ISBN as well as the homework validating the 13 digit ISBN. Other than those 2 sources I did not receive any substantial assistance with this project. The only help I did receive was either by reading the book or by reading forums online. Once I decided to forgo using the Amazon Product Advertising API, I required almost no aid.